

# 유닉스 웹 클러스터 시스템 환경에서 멀티미디어 데이터의 읽기 성능 개선방안

김영애\*, 이혁\*\*, 최진영\*\*

\*고려대학교 컴퓨터공학과

\*\*고려대학교 컴퓨터학과

e-mail : clspast@korea.ac.kr , \*\*{hlee, choi}@formal.korea.ac.kr

## A method for enhancing reading performance of multimedia data in Unix web cluster

Young-Ae Kim\*, Hyuk Lee\*\*, Jin-Young Choi\*\*

\*Dept. of Computer Engineering, Korea University

\*\*Dept. of Computer science, Korea University

### 요 약

최근 들어 더욱 UCC(User Created Contents)등과 같은 대용량 멀티미디어(multimedia) 서비스에 대한 요구가 나날이 증가되면서 부하분산에 중점을 둔 웹 클러스터링 시스템(Web Clustering System)에서 기존의 작은 크기의 스트림 데이터(Stream Data)나 조금 더 다양한 데이터를 위한 읽기 성능을 대용량 데이터에 초점을 맞춘 방안으로 최적화 시키는 것이 중요시되고 있다.

본 논문에서는 대용량 멀티미디어를 중심으로 실제 서비스시 간과 되어질 수 있는 운영체제(Operating System, O/S)에서의 I/O 인식, 디스크 제어 프로그램에서의 I/O, 웹 클러스터의 부하분산정책의 파라미터(Parameter)를 개선함으로써 읽기성능 향상 방안을 제시한다.

### 1. 서론

최근 인터넷 사용자의 급속한 증가와 다양한 웹 서비스의 등장으로 인터넷을 통해 송수신 되는 데이터의 멀티미디어화에 따라 네트워크 트래픽이 급증하고 병목 현상화 되고 있다. 이에 대한 해결책으로 얼마 전까지는 백본(Backbone)의 대역폭 확장에 주력하였으나, 이제는 서버의 수용한계를 넘는 트래픽에 대한 해결책으로 서버의 성능향상과 가용성을 확보하는 방안이 중요시 되고 있다[1]. 이를 위해 서버들을 고속 네트워크로 연결하고 이 서버들에 부하분산 기법을 적용하여 고성능 컴퓨팅을 제공하는 클러스터 시스템 구축이 증가하고 있다[2].

클러스터 시스템(Cluster System)은 단일 시스템으로 관리되는 독립적인 서버 그룹으로 여러 대의 컴퓨터를 서로 연결하여 마치 하나의 컴퓨터처럼 사용하는 것이다[3]. 따라서 사용자와 시스템 관리자에게 개별 컴퓨터의 그룹이 아니라 하나의 시스템처럼 보인다[4].

클러스터 시스템은 사용목적에 따라 과학계산용과 부하분산용으로 구분할 수 있는데, 부하분산이 목적인

클러스터는 단일서버를 사용할 때 발생할 수 있는 부하를 여러 서버로 분산시켜 웹 서버의 과부하를 해결한다[5].

또한 클러스터는 장애 발생 시에도 중단 없는 서비스를 제공할 수 있도록 이중화된 하드웨어와 소프트웨어 구성요소를 갖춘 시스템의 그룹이다[6].

이렇게 장애 대처와 웹 서버의 성능을 위해 클러스터 시스템을 도입하는 것이 근간의 웹 서버구축의 추세이며, 또한 웹 서버를 클러스터 형태로 구성하였을 때 기존 웹 서버에서 취약한 확장성, 가용성, 경제적인 효율성을 만족시킨다[7]. 최근에는 단순한 불특정 사이즈의 단순한 text 파일요구 보다는 대용량 multimedia 데이터 요구에 따른 UCC, 드라마, 영화와 같은 웹 서버의 데이터 read 작업에 대한 성능 요구가 증가되고 있다.

본 연구에서는 운영체제의 데이터 인식 최적화를 위한 chunk size 조정, 대용량 multimedia에 적합한 stripe unit 사이즈, 클러스터 서버당 적합한 load 분배를 제시하여 Web 서버상의 Read 요구에 대한 성능 최적화 방안을 제공하고자 한다.

본 연구의 구성은 다음과 같다. 제 2 장에서는

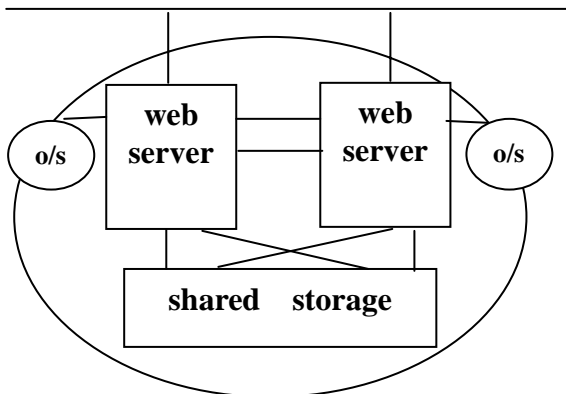
scalable web 클러스터의 이론적 배경과 문제점을 소개한다. 제 3 장에서는 read 성능개선을 위한 튜닝(Tuning) 관점을 소개한다. 제 4 장에서는 테스트를 통한 실증연구 및 결과를 분석한다. 마지막으로 제 5 장에서는 결과와 연구의 시사점 및 향후 연구 과제를 기술한다.

## 2. 관련 연구

### 2.1 scalable web 서버 클러스터의 개요

클러스터 web 서버는 최소 2 개의 노드(node)와 공유되는 디스크를 연결하고 두 node 간의 정보를 전달하는 기가비트 이더넷(Gigabit Ethernet) 연결로 이루어진다. 이러한 구성을 기준으로 그간 웹 서버의 성능에 미치는 요소를 정의하거나, 다양한 벤치마크를 통해 웹 서버의 성능을 측정하는 방법들에 관한 연구들이 이루어져 왔다[8]. 클라이언트(Client)에 Agent를 심어 서버의 응답시간을 측정하거나 ping, trace route 등의 명령을 이용하여 node 사이의 네트워크의 성능을 분석하게 하는 등 여러 가지 방식을 제시하고 있다[9].

제안하는 scalable web 클러스터 시스템은 (그림 1)과 같이 구성하였다.



(그림 1) scalable 웹 클러스터 구성도

### 2.2 scalable web 클러스터의 client 요청처리 특징

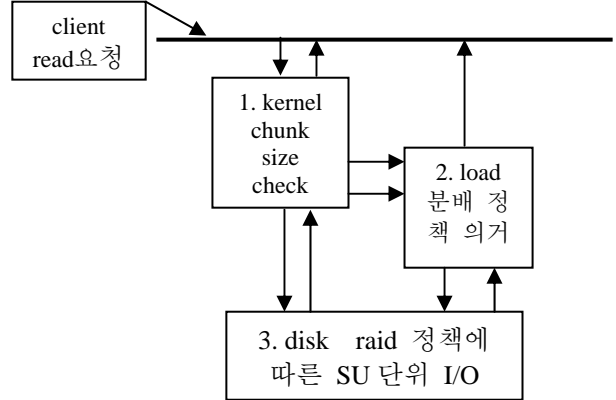
scalable web cluster 환경은 여러 대의 client 요청을 하나의 primary node 에서 받아들이며 해당 primary node 에서 load 분산정책에 의거 cluster group 내의 다른 node 들에게 client 의 요청을 분배한다. 그 후에 각 node 의 역할이 끝나면 해당 node 를 통해 직접 output 서비스가 이루어 진다.

클라이언트 요청은 다음 2 가지 경우로 나누어 분산 처리 작업이 들어가진다.

첫째는 client 당 요구 인식으로 같은 클라이언트가 여러 요구를 하더라도 하나의 요구로 인식하는 것이며, 둘째는 TCP connection 당 요구 인식으로 같은 클라이언트가 여러 요구를 하더라도 각각의 요구로 인식하는 것이다. 보통의 경우 클라이언트의 요청은 client 당 하나의 요청으로 웹 서버에서 load 분배하

도록 구성한다. 본 연구에서도 client 당 요구 인식에 대한 load 분산을 위주로 하며, 어느 정도의 load 분산정책이 read 서비스 시간을 단축시키는지 측정하여 해당 load 분산 정책 또한 서비스와 성능에 영향을 미치게 된다는 제안을 확인하고자 한다.

### 2.3 scalable web 클러스터 Data 접근 Interface



(그림 2) scalable 웹 클러스터 Data 접근 Interface

scalable web 클러스터 Data 접근 Interface 방식은 (그림 2)와 같이, 가장 먼저 kernel 에서 인식하고 있는 chunk size 에 의거 data I/O 단위를 측정 한 후에 각 cluster node 가 공유하고 있는 디스크의 RAID(Redundant Array of Inexpensive Disk) 정책에 따른 SU(Stripe Unit) 에 의거하여 read 작업을 하게 된다. 보통 Disk Raid 구현 시 기본값을 설정하는 구성 방안에 착안하여 본 연구에서는 700mb 이상의 대용량 Multimedia data 로 실험 측정 후 가장 적절한 SU size 를 제시하고자 한다. 본 연구에서는 Data redundancy 를 고려한 Raid 1 을 기준으로 작업 하였다.

### 2.4 scalable web 클러스터의 Tuning 개념 및 절차

튜닝이란 OS, CPU, Memory, Disk, I/O Controller, BUS, Disk Controller, Network 등을 이해하고 점검하여[10], Transaction 실행 시간을 향상시키는 일련의 작업을 말한다.

갈수록 복잡화, 대량화되고 있는 시스템을 그대로 유지하면서 read 요청에 대한 User Response Time 의 실행 시간을 단축 시키는 Tuning 은 투자 대비 이익률(ROI : Return On Investment)에 탁월한 효과를 가져온다는 점에서 중요시 되고 있다. 또한 튜닝을 보다 체계적으로 실시하기 위해서는 적절한 Tuning Scenario 가 필요하며[11] 본 논문에서는 다음과 같은 <표 1> Scenario 를 바탕으로 튜닝을 실시 하였다.

<표 1> Tuning 절차[11]

서버 튜닝	Kernel 단의 I/O 인식단위튜닝
	Cluster Load 분배 처리 튜닝
I/O 튜닝	데이터 I/O 단위 Stripe-Unit 튜닝

### 3. Read 성능개선 제안

본 논문에서는 기존 방식들에서 채택하고 있는 경험적 측면에 의한 웹 클러스터의 대용량 데이터 read 설정에 대해 3 단계로 나누어서 정확한 read 성능측정을 제안하고 적용하고자 한다.

#### 3.1 Kernel 단의 I/O 인식 파라미터 튜닝

UNIX 에서는 kernel 의 File system 파라미터 중에 I/O 작업 단위의 기본이 되는 maxcontig 가 있으며 기본값이 16 이다. 이 값의 적용은 16 \* 8K (메모리에서 인식하는 기본 페이지 단위) = 128 K 단위의 값으로 커널에서 한번 I/O 시에 인식하는 최대 값이므로 대용량 멀티미디어를 위한 값으로 해당 값을 최대 값인 128 로 늘려서 총 128 \* 8K = 1024K 만큼의 1 회 I/O 인식단위로 제안한다.

#### 3.2 Cluster Load 분배 처리 튜닝

Scalable Cluster 의 load 분배는 전적으로 clustering 구성 시에 primary node 와 다른 node 로 구분하여 설정하는데 기본적인 값은 primary node 에 2 개의 client 요청 처리, 다른 node 에 3 개의 client 요청 처리를 기준으로 설정하므로 본 논문에서는 대용량 데이터처리를 위해 primary node 에서 3 개와 2<sup>nd</sup> node 에서 5 개의 read 요청을 처리하도록 제안한다. Cluster 구성문 안에 scalable 처리 분배를 위한 설정에 3@1,5@2 선언을 한다.

#### 3.3 데이터 I/O 단위 Stripe-Unit 튜닝 적용

공유 디스크상의 동일 데이터 요구가 많은 환경에서는 Read data 를 2 개 이상의 공간에 저장하는 Raid 1 을 적용하는 것이 효율적이다. 그러나 그간 Raid 1 의 Stripe Unit 값은 64K 정도의 값을 사용해오는 것이 통상적이었다. 본 논문에서는 Raid 구성에서 인식할 수 있는 최대값은 1024K 를 최대한으로 맞출 수 있도록 6 개의 디스크를 Raid 1 기법에 적용시켰다.

## 4. 성능 실증 분석

### 4.1 환경의 구성

본 연구에서는 1 set 으로 구성된 2 node Cluster Web Server, 각 node 와 연결된 공유 디스크 환경에서 테스트를 진행 하였으며 각 Web Server 는 100Mbps Ethernet 으로 Heart-bit 및 Load 분배 역할을 하게끔 구성 하였다. Server 의 H/W 내역과 S/W 구성은 <표 2> 와 같다.

<표 2> 실험 H/W 및 S/W 사양

	Web Server	RAID 지원	디스크
H/W	SUN Sunfire 480 * 2	-	SUN T3 array

S/W	Sun Cluster 3.1 Apache 1 SUNOS 5.9	Veritas Volume Manager 4.0	-
HDD	17G	-	816GB

본 논문의 Tuning 대상 목록은 <표 3>과 같다.

<표 3> Tuning 대상 목록

	Kernel Chunk Size	Cluster Load 분배	Disk SU
기본	16	2@1,3@2	32k

### 4.2 결과 및 분석

아래의 결과는 Cluster Web Server 의 Tuning 전·후 30 번의 대용량 multimedia data 요청 건에 대한 Kernel 단의 I/O 인식 단위 튜닝, Cluster Load 분배 처리 튜닝, 데이터 I/O 단위 Stripe-Unit 튜닝을 측정한 평균 값 들이다. 마지막에는 각 단계별 측정을 모두 적용한 최종 값을 표현한다.

#### (1) Kernel 단의 I/O 인식단위 튜닝

Read 시간에 대한 서비스 속도 향상은 Tuning 전에 비해 제안 값을 설정한 결과 0.46 초 정도의 향상이 나타나게 되었다. 그 결과는 <표 4>와 같다.

<표 4> Kernel 단의 maxcontig 튜닝 (단위: 초)

	Tuning 전	Tuning 후	개선 효과
Read 시간	3.79	3.43	9.5 %

Cluster 의 실제 제어를 담당하는 kernel 에서의 I/O 인식단위인 chunk size 를 변경 함으로서 위와 같은 read 성능 향상을 확인할 수 있었다.

#### (2) Cluster Load 분배 처리 튜닝

<표 5>는 Web Server 의 Read 요청에 대한 Load 분배 후 속도에 대한 측정이다. 기존 평균 2 개의 요청을 primary node 가 처리하고, 3 개의 요청을 2<sup>nd</sup> node 가 처리하는 것에 비해 약 6.33 % 정도의 서비스 속도 향상을 보였다. 이는 Tuning 시 설정된 cluster node 당 처리하는 요청에 대한 설정 값을 통해 Access 하는 경우, 일상적으로 설정된 primary node 의 부하율이 감소 되었기 때문이다.

<표 5> Cluster Load 분배 처리 튜닝 (단위: 초)

	Tuning 전	Tuning 후	개선 효과
Load 분배에 따른 서비스 속도	3.79	3.55	6.33 %

#### (3) 데이터 I/O 단위 Stripe-Unit 튜닝

<표 6>은 Disk 데이터의 I/O 단위인 Stripe-Unit 튜

닝 측정 현황으로 3.79 로 측정된 평균 read service time 이 3.37 로 11.1%의 성능 개선 효과를 보였다. 이는, Raid 1 을 적용하여 read 요청이 많을 경우 동일 read 정보를 가진 구성을 유지하며, 많은 양의 data 를 read 함에 있어 SU 값을 조절하는 것이 유용함을 보여준다.

< 표 6> 데이터 I/O 단위 Stripe-Unit 튜닝 측정현황 (단위: 초)

	Tuning 전	Tuning 후	개선 효과
Read 속도	3.79	3.37	11.1 %

(4) 제시한 모든 조건을 적용한 튜닝 측정 현황

<표 7>은 실험한 모든 조건을 적용한 결과 값으로 평균 3.79 초로 측정된 기존 값에 비해서 각각의 개선 효과의 시너지 발생으로 3.43 초의 결과를 비교함으로써 9.49 %의 read 속도 개선이 증명되었다.

<표 7> 모든 개선안을 적용한 튜닝 (단위: 초)

	Tuning 전	Tuning 후	개선 효과
Read 시간	3.79	3.43	9.49 %

## 5. 결론 및 향후 연구방향

본 연구의 목적은 대용량 multimedia read 서비스 시간을 향상시키는 성능개선기법을 UNIX web cluster System 환경에서 disk 구성 위주로 실증적인 검증을 해 내는 것이다. multimedia read 서비스 시간을 향상을 위하여 본 연구는 cluster 서비스 구성 시에 디스크 Raid 1 구성 최적화 및 클러스터의 read 요청분배 구성에 대한 효율적인 가이드라인에 따르지 않고 기존의 관습적인 수치에 따른 관행에 의해 구성된다. 관점에서 문제에 접근하였고, 이를 검증하기 위해 700mb 이상의 사이즈를 가지고 있는 총 30 개의 multimedia data 를 대상으로 분석을 실시 하였다.

분석 환경은 2 개의 공유 storage 를 소유하는 2 node 의 Web cluster Server 를 구축하였으며, 연구 결과를 요약하면 다음과 같다.

Kernel 단의 I/O 인식단위튜닝, Cluster Load 분배 처리 튜닝, 데이터 I/O 단위 Stripe-Unit 튜닝을 제시함으로써 read 요구에 대한 서비스 성능을 전체적으로 9.49 % 개선하였다.

본 연구에서 제안하고 있는 3 가지는 다음과 같다. 첫째, 클러스터 O/S 자체가 지원하는 Chunk Size check 하여 서비스를 제어하는 kernel 단에서도 data size 에 맞는 효율적으로 적합한 I/O 인식을 할 수 있도록 하여 성능을 향상시킬 수 있다. 둘째, 대용량 700MB 기준으로 적합한 디스크당 I/O 단위인 SU size 를 적합하게 조절함으로써 성능을 향상시킬 수 있다.

셋째, scalable cluster 환경에서의 web 서버의 read 요청에 대한 load 분산을 적절히 제시함으로써 성능을 향상시킬 수 있다. 본 연구에서 제시한 load 분배는 많은 요청이 한꺼번에 몰릴수록 primary node 의 부하를 기존에 비해 줄임으로 더욱 속도 개선 효과가 있음을 알게 되었다.

여러 경우의 유저요구를 위한 다양한 size 의 data 를 가지고 있는 web server 의 경우, 제시된 Tuning 을 통하여 모든 read 의 성능을 향상시키는 것은 불가능하며 또한 클러스터환경에서 제시하는 장애 대처 환경을 고려하여 설계된 O/S 및 디스크의 특정 자원에 국한된 Tuning 을 실시 하였다.

향후 관련 연구는 상기와 같은 본 연구의 한계점을 극복하기 위하여 다양한 o/s server 환경에서의 성능 개선을 통하여 대용량 multimedia 의 read 서비스 시간을 보다 줄일 수 있는 성능 개선 방법론을 제시 하는 것이다.

## 참고문헌

- [1] Wensong Jhang, The paper "Linux Virtual Servers for Scalable Network Services" Ottawa Linux symposium, 2000
- [2] Jian liu, Lorghu Xu, Baogen Gu, Jing Zhang, "A Scalable, high performance cluster server" High Performance computing in the Asia-Pacific region, 2000. Proceedings. The firth International Conference/Exhibition, Vol.2 pp941-944, 2000
- [3] G.Hunt, E.Nahum, and J.Tracey. Enabling Content-based Load Distribution for Scalable Services. Technical report, IBM T.J. Watson Research Center, May 1997.
- [4] 정지영, "웹 서버 클러스터를 위한 효율적인 내용 기반의 부하 분배" pp1, 2004.
- [5] 권오영, "클러스터 시스템 개요" 기술·정책 동향, 한국과학기술정보연구원 소식지, 2000.
- [6] 최우영, "유닉스 클러스터 시스템의 서비스 인계시간 단축방안" pp12, 2005.
- [7] 윤천균, "웹 클러스터 시스템의 실시간 서버 상태를 기반으로 한 부하분산 방안" pp1-2, 2005.
- [8] Wimon ho, WayneM.Loucks and Ajit Singh "Monitoring the Performance of a Web Service", IEEE, 1998.
- [9] L Cottrell and C Logg. "Tutorial on WAN Monitoring at SLAC", <http://www.slac.stanford.edu/comp/net/wanmon/tutorial.html>, March, 1997.
- [10] 이원조, "관계형 데이터베이스에서 셀프 조인 쿼리를 위한 성능 평가" 2001.
- [11] 김광은, "관계형 데이터베이스 시스템 성능 향상을 위한 튜닝 방안" pp12, 24, 25, 2003.